Integrated Circuit Analysis on Solid-State Drive

Dhanika Novlisariyanti - 13521132[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13521132@std.stei.itb.ac.id

***Abstract*— Not just theoretical, but applied in real life, graph is very useful. Graph implementation can be found anywhere, such as network connections, city planning, analog circuits and etc. The focus on this journal will be integrated circuit in SSD. As SSD have become an tempting alternative to HDDs, it peaks interest into people. SSDs are increasingly applied in high performance computing devices. Computing devices have been a great help towards human mankind, so this journal will peek through how SSDs internal architecture are and how integrated circuit played an important part.**

***Keywords*— Integrated Circuit, SSD, Planar Graph.**

I. INTRODUCTION

Imagining the current world right now without electronics is impossible. Electronics have become a daily use by civilization with how amazing their performance is by helping some daily tasks. Other benefits include how efficient it is to task and less time consuming. Electronics have come a long way and is constantly upgraded to be better, so it matches with daily life problems. One of them is computer. Progressively, computer has been upgraded constantly so it's lighter in terms of weight, faster in its operation time. Computer is integrated by many components, such as processor, motherboard, storage, memory, and etc. As for now, the one main point this journal will analyze is its Solid-State Drive or SSD. SSD is a solid-state storage device that uses integrated circuit assemblies as memory to store data persistently. Unlike hard disk drive or HDD, it does not contain spinning disks and moveable read/write heads. SSDs have quicker access time and lower latency. However, SSD contains data when there are electrical charges. If the computer is shut down, then the data will be lost. SSDs implement one of the subjects in discrete mathematics, and that is graph planar. A graph is called planar if all edges in the given graph do not cross. This is important in SSDs because if the wired circuit cross path each other, it can lead into electrical malfunction which engineers do not want. Specifically, SSD used a controller which use the NAND logic gate.

II. BASIC THEORY

A. Graph

Graph is defined as a set of V and E, denoted as (V, E), with V as a set of vertex that is not empty and E as a set of edges that can be empty or not. Graph is classified whether it has a loop or a double-sided edge on the graph. Simple graph is defined as graph that does not have a loop and double sided edge and unsimple graph is defined as a graph that contains a double sided edge or loop. In theory, there are many other types of graphs. One of them is planar graph. Planar graph is defined as graph that can be drawn in a plane such that so no edges cross each other. Planar graph representation splits the plane into connected areas. For each graph, there is a connection between the number of vertices(v), the number of edges(e) and the number of area(f). This is called Euler's formula.

$$n - e + f = 2$$

Not all graphs are planar. And some planar graph have different areas when it is drawn as a planar graph or other graphs that is isomorphic to the corresponding graph. The planarity on graph can be determined with Kuratowski's Theorem or with the Euler's formula. This is proven by K_5 graph, which is shown below.

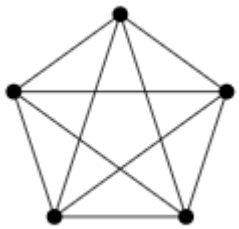

Figure 1 Graph K5

Source: discrete.openmathbooks.org

Graph K_5 has 5 vertices and 10 edges, from that we get 7 areas. Each area must be surrounded by at least 3 edges. The total number of boundaries around all faces in the graph must be greater than 3f. But since the total number of boundaries is 2e, it is impossible, since it has been determined that f = 7 and e = 10, and $21 \leq 20$. This is a contradiction to the Euler's formula so K_5 is not planar.

Another graph that is not planar is $K_{3,3}$.

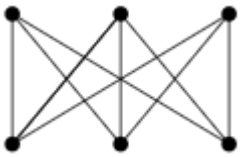

Figure 2 Graph $K_{3,3}$

Source: discrete.openmathbooks.org

Assume $K_{3,3}$ is planar. By using Euler's formula, there will be 5 area, since there are 6 vertices and 9 edges. Since each edge is used as a boundary twice, $4f \leq 2e$. But from that it gets $20 \leq 18$, which is false. Thus, $K_{3,3}$ is not planar.

Kuratowski's Theorem said that a graph **G** is not planar if and only if **G** has a subgraph that's a subdivision of $K_{3,3}$ and K_5 [1]. Any graph can be determined its planarity with this theorem, the only hard catch is actually finding the desired subgraph. It usually uses some trial and error method to find the subgraph. But it is important to remember that for **G** to have a subgraph that's a subdivision of either $K_{3,3}$ or K_5 , it still has to have the same number of vertices.

B. Integrated Circuits

Integrated Circuits is built from components such as resistors, transistors, capacitors, etc. All of those are connected to achieve a goal. Integrated Circuits is usually stuffed into chips. Those chops are layered by semiconductor wafers, copper, and other materials which interconnect to form transistors, resistors or any other components inside a circuit. These are called a **die**.

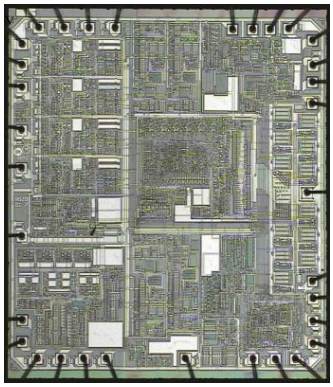

Figure 3 Integrated Circuit

Source: https://learn.sparkfun.com/

Those semiconductor and layers of coppers are incredibly thin. It is built to be in its smallest possible form. A typical integrated circuit design involves several steps, such as system specification, architectural or system level design, logic design, circuit design, physical design, physical verification and signoff, layout post processing, fabrication, packaging, and testing.

Logic gates in integrated circuits is equal to the vertices in graph while the semiconductors represent the graph edges. The edges between each vertex are weighted according to the maximum statistical cross correlation value observed in the signals traveling between the vertices [2]. During the IC placement process, all standard cells or gates are placed in legal locations on-site rows. This is to minimize the wire length while ensuring optimal placement that will help faster time convergence. Logic gates itself can contains its own integrated circuit. Some logic gates usually contain a lot of gates in one package. These logic gates are connected inside an integrated circuit to create timers, counters, latches, shift registers, and other basic logic operations.

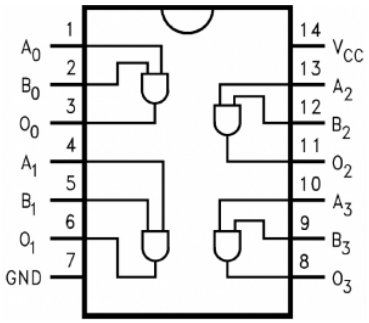

Figure 4 Logic Gates

Source: https://learn.sparkfun.com/

C. Solid State-Drive

A solid-state drive (SSD) is a storage technology, based on flash memory that in some situations is an attractive alternative to the conventional rotating disk [3]. Reading from SSDs is usually faster than writing. The difference between random reading and writing performance is caused by a fundamental property of the underlying flash memory. The important components in SSDs are firmware, flash memory, and controller. Flash memory has a few characteristics, such as the unit for read and write operations. This is what affects the speeds of various operations. Then the same physical page in the memory can be written upon just once after each erase operation, and each block of it has a limited number of erase cycles. As those blocks repeatedly rewrite it can eventually wear out leading to SSDs not functioning well. Thus, the method for decreasing the number of write operations is very important for SSDs. If the number of write operations is reduced, the life-time of an SSD is extended and the input (I/O) processing speed is improved. The read and write operations on SSDs have been improved throughout the year. From all those trials and errors, flash bashed SSD catching is the best solution for boosting today's storage system. The old method consists of using two-level cache composed of RAM and a flash memory secondary cache. The more modern method divided the read and write operations into different isolation process. Read process is usually faster than write, it's due to the effect of NAND memory placing that cannot be overwritten with a single IO operation.

An SSD contains multiple NAND flash chips for storing data. Each chip contains one or more dies, and each die contains one or more planes. A plane is divided into blocks, and a block is divided into pages [4]. Recently, most NAND flash relied on floating gate technologies, when electrons are trapped between two oxide layers in a region called the floating gate. Each time voltage is applied, and electrons passed through the oxide layer, the layer degrades slightly. Continuous write and erase operations, will affect the degradation and eventually the cell might no longer be viable.

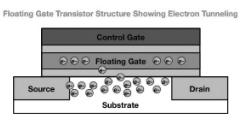

Figure 5 NAND Memory Cell

Source: embeddedts.com

III. SOLID-STATE DRIVE CIRCUIT ANALYSIS

A. INTEGRATED CIRCUIT METHOD

Designing graph with large numbers of vertices and vertexes is tricky. Condensing graph is an essential tool in identifying logic centers withing a layout. This method is based on vertex importance or vertex similarity. This importance method uses vertices weight functions based on degree and shortest path in a graph. Such weight functions favor strongly connected vertices. This result in producing a graph using only the higher weight vertices.

Logic gates withing a layout equate to the vertices in the graph while the conductive connections between the logic gates represent the graph edges. The edges between each vertex are weighted according to the maximum statistical cross-correlation value observed in the signals traveling between the vertices [6]. To get this, there are a few algorithms including graph vertex importance, condensation, cycle counting, partitioning, and associated time and class computing complexities. The first method condensed and strongly connected graph achieve by compress merging vertices with minimum number of common neighbor. This is effective for locating component centers. The next one vertex centrality is split into two categories degree-based and geodesic. Degree based algorithm uses the degree of a vertex and the degree of its neighbors while geodesic algorithms use computation of shortest path [6].

The neighborhood-based Centre originally roots from the concept of node's influence is correlate to its capacity to impact the behavior of its surrounding neighbors. Nodes with a similar number of neighbors are distinguished by the clustering coefficient. Those nodes have greater influence or importance than large ones. When considering a node's neighborhood-based centrality, the location of a node also determines its importance. Path-based centrality or geodesic, calculate the shortest path between all vertices in a graph. The shortest distance between nodes is considered more centric and have a higher importance.

B. SOLID STATE DRIVE ARCHITECTURES

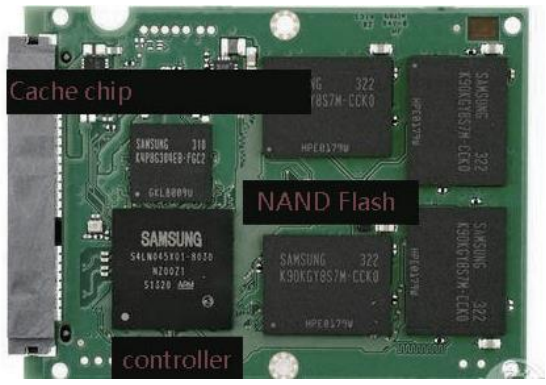

Figure 6 SSD Internals (Samsung 840EVO) [10]

It is known that SSDs re made entirely of electronic components, there are no moving or mechanical parts. Voltages applied to the floating gate transistors that will be read, written, and erased. This wiring transistors will either use the NOR flash memory or the NAND flash memory. But it has been scientifically proved that NAND flash memory is better. From chapter II, it has already been mentioned that NAND-flash are wearing off and have limited lifespan. Diving more how it can be worn out, based on Figure 5, electrons might get trapped in the transistor by mistake, and through time the amount of electrons that can be trapped will increased leading to the cells worn out. There are three types of cells in the industry and that is single level cell (SLC), multiple level cell (MLC), and triple level cell (TLC). On single level cell, transistors can store only 1 bit but have a long lifespan. On multiple level cell, transistors can store 2 bits, with higher latency and reduced lifespan. Then the triple level cell, can store 3 bits with even higher latency and reduced lifespan [8].

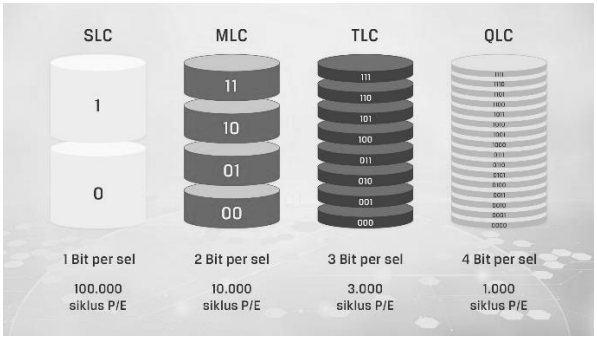

Figure 7 SLC, MLC, TLC

The internal parallelism makes a single SSD capable of handling multiple data. The mapping policy directly determines the data layout across flash memory packages, which affects the efficiency of parallelizing data access. A host interface logic connects the host through and interface connection, the IDE bus. An SSD controller manages flash memory space, translates incoming request, and issues commands to flash memory packages via a flash memory controller [5]. The internal architecture of SSDs shows the parallelism available at different levels, and operations at each level to be parallelized.

These parallelisms have different level such as channel-level parallelism, package-level parallelism, chip-level parallelism, and plane-level parallelism. By combining all the levels of internal parallelism inside an SSD, multiple blocks can be accessed simultaneously.

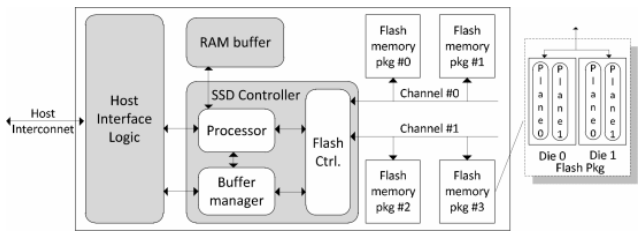

Figure 8 An Illustration of SSD architecture [5]

Obtaining information about architectural information is hard for several reasons because some designs are often copyrighted or is an intellectual property to some manufacturers. Based on the experimental approaches to infer directly the three

key factors, this is some speculation about the internal architecture and polices adopted in the SSD.

A domain is a set of flash memories that share a specific set of resources like channels. [5] A chunk is a unit of data that is continuously allocated within one domain [5].

```
init_SSD():  sequentially write SSD w/ 256KB req.
rand_pos(A): get a random offset aligned to A sect.
read(P, S):  read S sectors at offset P sect.
stride_read(J,D): read 1 chunk with J jobs from
          offset 0, each read skips over D chunks
plot(X,Y,C): plot a point at (X,Y) for curve C
M:  an estimated max. possible chunk size
D:  an estimated max. possible interleaving degree

(I) detecting chunk size:
  init_SSD();              // initialize SSD space
  for (n = 1 sector; n <= M; n *= 2): //req. size
    for (k = 0 sector; k <= 2*M; k ++):// offset
      for (i = 0, latency=0; i < 100000; i ++):
         pos = rand_pos(M) + k;
         latency += read (pos, n);
      plot (k, latency/100000, n); //plot avg. lat.

(II) detecting interleaving degree:
  init_SSD();              // initialize SSD space
  for (j=2; j <=4; j*=2):  //  num. of jobs
    for (d = 1 chunk; d < 4*D; d ++)://stride dist.
       bw = stride_read (j, d);
       plot (d, bw, j);          //plot bandwidth
```

Figure 9 Pseudocode of uncovering SSD internals [5]

It has been mentioned earlier that as through time solid state drive can be worn out. This is still a legitimate concern towards SSD reliability. Redundant array of independent disks (RAID) provides an option to improve reliability of SSDs. This is another dive in into solid state drive architecture, as their architecture is private information from their manufacturers'.

Specific Notations of SSD		
M	:	Erasure limit of each block (e.g., 10K)
B	:	Total number of blocks in each SSD
$\lambda_i(t)$	:	Error rate of a chunk in SSD i at time t
Specific Notations of RAID Array		
N	:	Number of data drives (i.e., an array has $N+1$ SSDs)
S	:	Total number of stripes in an SSD RAID array
p_i	:	Fraction of parity chunks in SSD i, and $\sum_{i=0}^{N} p_i = 1$
k	:	Total number of erasures performed on SSD RAID array (i.e., system age of the array)
k_i	:	Number of erasures performed on each block of SSD i (i.e., age of SSD i)
T	:	Average inter-arrival time of two consecutive erase operations on SSD RAID array
$\pi_j(t)$	:	Probability that the array has j stripes that contain exactly one erroneous chunk each, $(0 \le j \le S)$
$\pi_{S+1}(t)$	:	Probability that at least one stripe of the array contains more than one erroneous chunk, so $\sum_{j=0}^{S+1} \pi_j(t) = 1$
$R(t)$	:	Reliability at time t, i.e., probability that no data loss happens until time t, $R(t) = \sum_{j=0}^{S} \pi_j(t)$

Figure 10 SSD Raid Organization [7]

C. BASIC OPERATIONS IN SOLID STATE DRIVE

The organization of NAND-flash cells, it is not possible to read or write single cells individually. Memory is accessed with specific properties. A NAND-flash page can be written to only if it is in "free" state. When data is changed, the content is then transferred to an internal register, the data will be updated.

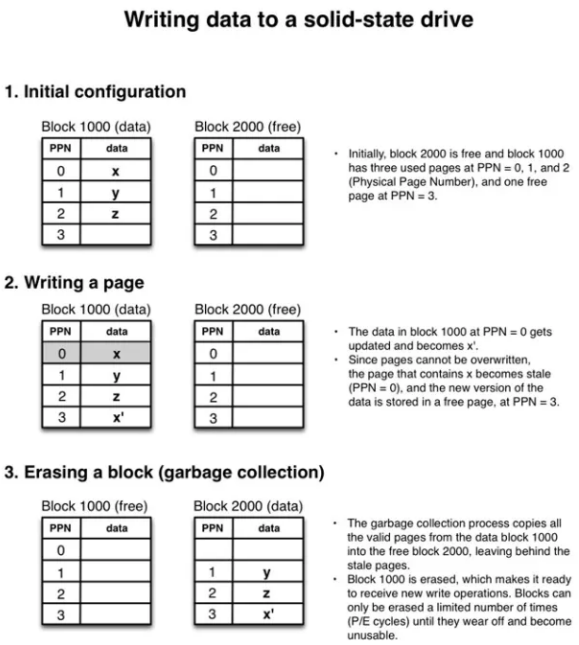

Figure 11 Read, Write, Erasing Operations

The wearing of NAND-flash cells can be prevented by the distribution work among cells as evenly as possible so that each blocks will reach their P/E cycle limit and wear of the same time. This is the task of FTL or Flash Translation Layer.

Flash Transition Layer resides in the SSD controller. It has two purposes which is logical block mapping and garbage collection. The logical block mapping translates logical block addresses from the host space into physical block addresses. Using page-level mapping to map any logical page from the host to a physical page is one of the approach. The log-block FTL allows for optimizations, the most notable being the switch-merge.

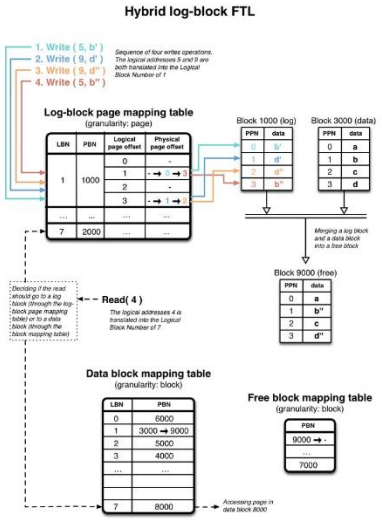

Figure 12 Log-Block FTL

The log-block mapping scheme has been constantly improved such as FAST, superblock mapping and etc. The most recent implementation is using the log-block mapping that allows random writes to be handled like sequential writes.

When writing pages, pages cannot be overwritten, if it must be updated the new version will write to a free page. Because of this some controllers implement a garbage collection process in the SSD so the incoming write commands can be processed.

Top optimize SSDs functionality, a write request or other basic operations is also aligned on the page size that will be written to a NAND-flash physical page directly. Partition alignment showed improved performances on SSDs significantly.

D. HOW INTEGRATED CIRCUIT AFFECTED SOLID STATE DRIVE

Integrated Circuit design really affected the structural architecture inside the solid-state drive. IC is designed differently for each type of SSD depending on what is the purpose and depending on the production cost. IC is designed as well as it be so that it is not overlapped by one another. If it does, it will create malfunction to the chip or SSDs. Surely, drawing an integrated circuit in a small components and a lot of gates will take a lot of time. These drawings are helped by a lot of algorithm and software. One of the proved theorems is by a couple of engineers from Germany, they proved that Every IC-planar graph with n vertices admits a straight-line drawing in $O(n^2)$ area which can be computed in O(n) [11].

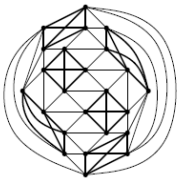

Figure 13 IC planar drawing

Another great effect using integrated circuit on SSD is its low impact on flash memory, in HDD the rotating disk or the magnet parts may result in data corruption to the drive internals. But using integrated circuits if there are any damage in the electrical system, it can fire up the entire solid-state drive. It is relatively smaller than HDDs to since it doesn't have rotating disk, instead SDD have integrated circuit and connector.

Figure 14 SSD and HDD [12]

IV. CONCLUSION

Accepting the world as it is, can be a double edge sword. As we continuously improve technology, it is better to not be satisfied as to what it is now. Computer technology is constantly improved towards the decades, and it is important to analysis how the design and functions have changed to improve quality. SSDs are constantly upgrading. The implementation of integrated circuit towards SSDs are really important but it can be different for every manufacturer. Since it is classified, generalize people can predict on how the architecture inside by reverse engineering. The design of integrated circuit is as important as other components because overlapping conductors in the circuit can cause electrical errors. As computer and other technology is design to be more practical and be carried everywhere, constant development has been developed thoroughly. This will need a lot of study towards the theorem of graph and how we can improve it as well as integrated circuit. Another good take is by applying nano technology in the production of integrated circuits, as nano particles is really small, it can help towards the development of very compact and practical SSD.

V. ACKNOWLEDGEMENT

All praise and thanks be to Allah, the One True Almighty God so that I can finish this work. This work is a requirement to fulfill Discrete Mathematics task. Sincere gratitude to Mrs. Fariska Zakhralativa Ruskanda, S.T., M.T. for teaching this semester. Thank you for all the materials and knowledge at edunex site because I can freely access it. I would like to show my gratitude towards other lecturers Dr. Nur Ulfa Maulidevi, S.T, M.Sc.and Dr. Ir. Rinaldi Munir, MT. I am still lacking at many aspects, but I hope to get improved as I continue to gain knowledge. And lastly, I want to thank myself for hanging in there, for always trying my best even though the outcome may not be the best. I truly appreciated the process through all of it.

REFERENCES

[1] Oscar Levin, *Planar Graph*, Retrieved December 11, 2022, 1:49, from https://discrete.openmathbooks.org/dmoi3/sec_planar.html

[2] JIMBLOM, *Integrated Circuits*, Retrieved December 11, 2022, 1:55, from https://learn.sparkfun.com/tutorials/integrated-circuits/all

[3] Bryant, R. E., & O'Hallaron, D. R. (2019). Memory Hierarchy. In Computer Systems: A programmer's perspective. essay, Pearson, ch 6.

[4] Robert Sheldon, Storage 101: Understanding the NAND Flash Solid State Drive, December 12, 2022, 16:53, from https://www.red-gate.com/simple-talk/databases/sql-server/database-administration-sql-server/storage-101-understanding-the-nand-flash-solid-state-drive/

[5] F. Chen, R. Lee and X. Zhang, "Essential roles of exploiting internal parallelism of flash memory based solid state drives in high-speed data processing," 2011 IEEE 17th International Symposium on High Performance Computer Architecture, 2011, pp. 266-277, doi: 10.1109/HPCA.2011.5749735.

[6] Bryant, R. E., & O'Hallaron, D. R. (2019). Memory Hierarchy. In Computer Systems: A programmer's perspective. essay, Pearson. C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.

[7] Y. Li, P. P. C. Lee and J. C. S. Lui, "Analysis of Reliability Dynamics of SSD RAID," in IEEE Transactions on Computers, vol. 65, no. 4, pp. 1131-1144, 1 April 2016, doi: 10.1109/TC.2014.2349505.

[8] Emmanuel Goossaert (February 12, 2014), Retrieved December 12, 19:53, from https://codecapsule.com/2014/02/12/coding-for-ssds-part-2-architecture-of-an-ssd-and-benchmarking/#ref

[9] Kingston Technology Corporation (April, 2021), Retrieved December 12, 19:57, from https://www.kingston.com/id/blog/pc-performance/difference-between-slc-mlc-tlc-3d-nand

[10] ELINFOR, *Overview of SSD Structure and Basic Working Principle*, Retrieved December 12, 2022, 20:11, from https://www.elinfor.com/knowledge/overview-of-ssd-structure-and-basic-working-principle1-p-11203
[11] Brandenburg, F. J., Didimo, W., Evans, W. S., Kindermann, P., Liotta, G., & Montecchiani, F. (2016). Recognizing and drawing IC-planar graphs. Theoretical Computer Science, 636, 1–16. https://doi.org/10.1016/j.tcs.2016.04.026
[12] Herwin Lab, *Apa itu SSD?*, Retrieved December 12, 2022, 21:36, from https://www.herwinlab.com/apa-itu-ssd/

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 12 Desember 2022

Dhanika Novlisariyanti NIM 13521132